INTRODUCTION TO PYTHON
Final Project

Spanish Football League

Statement

Football (European football) is one of the most popular entertainment media in Spain. On average, a LaLiga game is watched by more than 450,000 people. Not to mention the case of Barcelona and Real Madrid, which easily exceed one million spectators in each game. Now LaLiga has almost 100 years of history, and it leaves us with it a good load of data to work with.

You are provided with a dataset containing historical data from LaLiga. All official matches (that we have record) from 1928 up until 2021 are available in it. You will see the details of this dataset in section Data.

The project consists of two different parts:

- **LaLiga Data Analysis**. 10 exercises proposed for you to carry out in a Jupyter Notebook using the data provided.
- **"La Quiniela" game**. You are asked to develop a machine learning model that predicts the outcome of a matchday using historical data. Just like playing "La Quiniela", your model must predict for every match of the matchday whether home team will win (1), away team will win (2) or there will be a tie (X).

You are also provided with a repository with the base structure for this project.

Data

The data is provided as a SQLite3 database that is inside the repository. This database contains the following tables:

- `Matches`: All the matches played between seasons 1928-1929 and 2021-2022 with the date and score. Columns are *season*, *division*, *matchday*, *date*, *time*, *home_team*, *away_team*, *score*. Have in mind there is no time information for many of them and also that it contains matches still not played from current season.
- `Predictions`: The table for you to insert your predictions. It is initially empty. Columns are *season*, *timestamp*, *division*, *matchday*, *home_team*, *away_team*, *prediction*, *confidence*.

The data source is Transfermarkt, and it was scraped using Python's library BeautifulSoup4.

LaLiga Data Analysis

Write a Jupyter Notebook called `LaLigaDataAnalysis.ipynb` in folder `analysis/` with your answers to the following exercises/questions. Once you are done with it, export it to HTML and save the result in `reports/` folder from the repo. Make sure that your notebook can be rerun from scratch without any errors.

1. Is it true that the home team is more likely to win? Make a pie chart showing the result distribution (whether home team wins, visitor team wins, or there's a tie) of all matches in the data. Write in the plot the percentage of each category.

2. What are the top ten scoring teams of all time? What are the ten teams that concede the most? Make two bar plot charts showing each of them. Consider only matches played in 1st division.
   What have been the biggest wins? Of course, when we say *biggest* we mean those with the highest goal difference. Show the top ten of them in a table.

3. There has been a lot of discussion about how LaLiga's television rights have changed game schedules in the last years. Make a bar plot chart showing the number of matches played each weekday, and make also a histogram of match time. Compare this two graphics between seasons 2000-2001 and 2020-2021.

4. Build a cross results table for season 2020-2021 (1st division). Figure 1 is an example taken from Wikipedia. Try to make it the most similar to this one: use team abbreviations as column names and paint the background of each cell according to result (green in case local team wins and red in case visitor team wins). Also, could you model the intensity of this background color with the goal difference from the match?

Local \ Visitante	ALA	ATH	ATL	BAR	BET	CAD	CEL	EIB	ELC	GET	GRA	HUE	LEV	OSA	RMA	RSO	SEV	VAL	RVA	VIL
Deportivo Alavés	—	1–0	1–2	1–1	0–1	1–1	1–3	2–1	0–2	0–0	4–2	1–0	2–2	0–1	1–4	0–0	1–2	2–2	1–0	2–1
Athletic Club	0–0	—	2–1	2–3	4–0	0–1	0–2	1–1	1–0	5–1	2–1	2–0	2–0	2–2	0–1	0–1	2–1	1–1	2–2	1–1
Atlético de Madrid	1–0	2–1	—	1–0	2–0	4–0	2–2	5–0	3–1	1–0	6–1	2–0	0–2	2–1	1–1	2–1	2–0	3–1	2–0	0–0
F. C. Barcelona	5–1	2–1	0–0	—	5–2	1–1	1–2	1–1	3–0	5–2	1–2	4–1	1–0	4–0	1–3	2–1	1–1	2–2	1–0	4–0
Real Betis Balompié	3–2	0–0	1–1	2–3	—	1–0	2–1	0–2	3–1	1–0	2–1	1–0	2–0	1–0	2–3	0–3	1–1	2–2	2–0	1–1
Cádiz C. F.	3–1	0–4	2–4	2–1	0–1	—	0–0	1–0	1–3	0–2	1–1	2–1	2–2	0–2	0–3	0–1	1–3	2–1	0–0	0–0
R. C. Celta de Vigo	2–0	0–0	0–2	0–3	2–3	4–0	—	1–1	3–1	1–0	3–1	2–1	2–0	2–1	1–3	1–4	3–4	2–1	1–1	0–4
S. D. Eibar	3–0	1–2	1–2	0–1	1–1	0–2	0–0	—	0–1	0–0	2–0	1–1	0–1	0–0	1–3	0–1	0–2	0–0	1–1	1–3
Elche C. F.	0–2	2–0	0–1	0–2	1–1	1–1	1–1	1–0	—	1–3	0–1	0–0	1–0	2–2	1–1	0–3	2–1	2–1	1–1	2–2
Getafe C. F.	0–0	1–1	0–0	1–0	3–0	0–1	1–1	0–1	1–1	—	0–1	1–0	2–1	1–0	0–0	0–1	0–1	3–0	0–1	1–3
Granada C. F.	2–1	2–0	1–2	0–4	2–0	0–1	0–0	4–1	2–1	0–0	—	3–3	1–1	2–0	1–4	1–0	1–0	2–1	1–3	0–3
S. D. Huesca	1–0	1–0	0–0	0–1	0–2	0–2	3–4	1–1	3–1	0–2	3–2	—	1–1	0–0	1–2	1–0	0–1	0–0	2–2	0–0
Levante U. D.	1–1	1–1	1–1	3–3	4–3	2–2	1–1	2–1	1–1	3–0	2–2	0–2	—	0–1	0–2	2–1	0–1	1–0	2–2	1–5
C. A. Osasuna	1–1	1–0	1–3	0–2	0–2	3–2	2–0	2–1	2–0	0–0	3–1	1–1	1–3	—	0–0	0–1	0–2	3–1	0–0	1–3
Real Madrid C. F.	1–2	3–1	2–0	2–1	0–0	0–1	2–0	2–0	2–1	2–0	2–0	4–1	1–2	2–0	—	1–1	2–2	2–0	1–0	2–1
Real Sociedad	4–0	1–1	0–2	1–6	2–2	4–1	2–1	1–1	2–0	3–0	2–0	4–1	1–0	1–1	0–0	—	1–2	0–1	4–1	1–1
Sevilla F. C.	1–0	0–1	1–0	0–2	1–0	3–0	4–2	0–1	2–0	3–0	2–1	1–0	1–0	1–0	0–1	3–2	—	1–0	1–1	2–0
Valencia C. F.	1–1	2–2	0–1	2–3	0–2	1–1	2–0	4–1	1–0	2–2	2–1	1–1	4–2	1–1	4–1	2–2	0–1	—	3–0	2–1
Real Valladolid C. F.	0–2	2–1	1–2	0–3	1–1	1–1	1–1	1–2	2–2	2–1	1–2	1–3	1–1	3–2	0–1	1–1	1–1	0–1	—	0–2
Villarreal C. F.	3–1	1–1	0–2	1–2	1–2	2–1	2–4	2–1	0–0	1–0	2–2	1–1	2–1	1–2	1–1	1–1	4–0	2–1	2–0	—

Figure 1: Example of cross results table.

Write a function that, given the season and division, plots the cross results table. Function prototype should be like `plot_cross_results_table(season, division)` and return the plot object.

5. As you surely know, there has always been a historical rivalry between Barcelona and Real Madrid. But which of them has won the most games in direct confrontations? Which of them has scored the most goals in these games? Show both things in two pie charts, side by side. Remember to consider ties in the first one.
   Write a function that, given two team names, plots the two graphs described above. Function prototype should be like `plot_direct_confrontations_stats(team1, team2)` and return the plot object. Use it with some other classical rivals like Betis and Sevilla.

6. Between 1979 and 1980, Real Sociedad managed to chain a total of 38 games without losing. That was, by far, the longest undefeated streak in their history. Which teams have had the longest undefeated streaks? Show the longest undefeated streaks in a horizontal bar plot, indicating in each bar the team name and the dates it held that streak, for instance, `Real Sociedad 22/04/1979 - 04/05/1980`.

7. Create a table with the final standings of each season (and division), that is, a table that contains all the teams ordered (in descending order) by the number of points they got during that season, and some other aggregate statistics. The table must contain the following columns: `season`, `division`, `ranking`, `team`, `GF` (total goals scored), `GA` (total goals conceded), `GD` (goals difference), `W` (total wins), `L` (total loses), `T` (total ties), `Pts` (points). Remember that, in football, you earn 3 points per victory, and 1 point per tie (none for loses). In case two teams have same number of points, order by `GD` (descending), and then by `GF` (also descending). Order the table so that standings of one season come before standings of previous one, and standings of 1st division come before standings of 2nd division.

	season	division	rank	team	GF	GA	GD	W	L	T	Pts
0	2020-2021	1	1	Atlético Madrid	67	25	42	26	4	8	86
1	2020-2021	1	2	Real Madrid	67	28	39	25	4	9	84
2	2020-2021	1	3	Barcelona	85	38	47	24	7	7	79
3	2020-2021	1	4	Sevilla FC	53	33	20	24	9	5	77
4	2020-2021	1	5	Real Sociedad	59	38	21	17	10	11	62
...	...	...	...	...	...	...	...	...	...	...	...
2739	1928-1929	1	6	Athletic Madrid	43	41	2	8	8	2	26
2740	1928-1929	1	7	Espanyol	32	38	-6	7	7	4	25
2741	1928-1929	1	8	Catalunya	45	49	-4	6	8	4	22
2742	1928-1929	1	9	Real Unión	40	42	-2	5	11	2	17
2743	1928-1929	1	10	Racing	25	50	-25	3	12	3	12

Figure 2: Example of how standings table should look like.

   Save the final table in Excel with the name `SeasonStandings.xlsx` in the `reports/` folder.

8. Villarreal is a team that has grown a lot in recent decades. Specially ever since some billionaire guy bought it (Fernando Roig, from Mercadona).

Make a line plot showing the rank of Villarreal at the end of each season, from the oldest ones (left) to the earliest ones (right). Consider rankings in 2nd division to be a continuation of the 1st one, that is, if there's N teams in 1st division and Villarreal got r position in 2nd division, then it should be placed in $N + r$. Draw in the same plot a line showing the cut between 1st and 2nd division.
Write a function that, given n team names, plots the graph described above of each one of them superposed. Function prototype should be like `plot_ranking_evolution(team1, team2, ..., teamN)` and return the plot object (note that function should not take one array-type argument, but n arguments). Use it to compare the evolution of all the teams that currently play in 1st division.

9. In football jargon, those teams that are permanently descending and ascending between 1st and 2nd division are called *elevator teams*. What are the most *elevator teams* in LaLiga? Plot the history of the top 5 of them using the function from exercise 9.

10. Create a table that is the same as the one in exercise 7, but not only with the season final standings, but the standings at the end of each matchday. Columns are the same, including `matchday` that tells about which matchday from the season these standings are from. Would you be able to add a new column `last_5` with the result of last 5 matches? This column should contain a list like `["W", "L", "W", "T", "T"]`. In this list, the first item is the immediate previous match, the second one is the match before this one, and so on. If there are no 5 previous matches (because `matchday` < 6, for instance) then just make the list shorter.
Save the final table in Excel with the name `MatchdayStandings.xlsx` in the `reports/` folder.

"La Quiniela" game

In this part you'll build a machine learning model emulating a "Quiniela" game. You are asked to build a model that aims to predict the result of each match in a matchday – either local team wins (`1`), visitor team wins (`2`) or there is a tie (`X`) –. You are provided with a base skeleton for your project in this repo. See `README.md` file in the repo for more information.

- Use data provided to build proper features and train any model you feel like the best for the case. For instance, the ranking of both teams in the standings, their number of goals scored... Use your imagination.
- Use Jupyter Notebooks to play around with your model until it feels ok to you. Use all notebooks that you need, storing them in `analysis/` folder. Don't panick about your code style in those notebooks, they are not going to be considered when grading. You are requested to write a final notebook called `ModelAnalysis.ipynb`, and **this is the only one that is important** (meaning, is the only one that will be graded). This notebook must contain the training and evaluation of the final model: how good is it, where does it fail the most, what are the causes of its errors, what are the more important features and how they behave, etc. Also, export this notebook to HTML and place it in `reports/` folder.
- Then (and only then), once you are comfortable with the result, move your code to well-structured Python modules in `quiniela/` folder to convert your exploratory notebooks into a functional software. You can use the provided `QuinielaModel` and extend it, or make it your style. Add modules to `quiniela/` folder if you need so, add parameters to `settings.py`, add arguments to `cli.py`. Feel like home, this skeleton is just a base for you. The project is yours.
- You can use all the third-party libraries you want as long as they are available in the PyPI repositories (that is, as long as you can install them with `pip`) and you explicitly add them in `requirements.txt`.
- Needless to say, the model is not expected to get all the results right, far from it. Football is (luckily) highly unpredictable. If the model gets about 40%-50% right, you can already consider to be great.
- You must include the name of team members in the file `README.md` (name, surnames and NIU of every team member). If there's anything new in the way to execute the program, or anything that your consider important to remark, put it also in here.
- Of course, all the code must be written in Python.

Teams

You must work in teams of two or three members. The name, surname and NIU of all team members must be written in the `README.md` file of your project. You must deliver only one copy of your project per team.

Delivery

You must deliver a *zip* file containing the root folder of your project through Virtual Campus. This folder must **not** contain the `laliga.sqlite` database. Also, do not include the virtual environment in the *zip* file (your environment should be reproducible through the `requirements.txt` file). The total size of the *zip* file must not exceed 10 MB.

The deadline to submit is December 15, 2021.

Grading

As you already know, there are two main blocks in this assignment, and so works the grading method.

- 5 points come from Data Analysis exercises. 10 exercises, 0.5 points each.
- 5 points come from La Quiniela model. These are divided as:
  - The correct operation of the software. Both methods `train` and `predict` must work without any errors (2 points).
  - The quality of `ModelAnalysis.html` report. The clarity of the observed insights and the conclusions obtained from the model behaviour (1 point).
  - The quality of the model. Effectiveness of the model and originality in feature engineering (1 point).
  - The quality of the code. Neatness, modularity and order of the code (1 point).

The final grade will be the sum of this two parts.